EE-322 Reintroduction to MATLAB (rev 3)

The Basics ... MATLAB operates on numbers that have been arranged into vector and matrix form. For example, typing **i=[0.5,1,1.5,2]** at the prompt assigns the values 0.5,1,1.5, and 2 to the variable **i**. The following command will also work, **i=[0.5 1 1.5 2]**. These values can be measures of current taken at four different times. Right now **i** is a row vector. We could have input the values in the form of a column vector instead by typing ...

i=[0.5
1
1.5
2]

or by typing **i=[0.5; 1; 1.5; 2]**. Here a carriage return or a semicolon is used instead of a comma to separate values. Using rows or columns is irrelevant right now; you just have to follow the rules of linear algebra when you want to operate on the vectors.

Suppose you want to plot the current **i** versus time. If you type **plot(i)**, the graph may look right but what is actually happening is that the horizontal axis is not time; we haven't defined the variable time yet. The variable **i** is being graphed versus each elements place in the vector, i.e., 0.5 is 1^{st}, 1 is 2^{nd}, etc... So we can define the time variable **t** by typing **t=1:1:4** or **t=1:4**. The outer numbers define the bounds on the range of **t**. The inner number defines the spacing, so that **t** takes on the values 1, 2, 3, and 4. If the inner number were 0.5, **t** would take on values 1, 1.5, 2, 2.5, 3, 3.5, and 4. This is almost twice as many values. Now to plot, the linear algebra rules are important: the vectors for **i** and **t** must be of the same length or size. Type **plot(t,i)**. Also try **plot(i,t)**. What's the difference?

Note that for any MATLAB command, you can type **help commandname** for an explanation. Type **help plot,** as an example.

The following exercises help you learn how to manipulate vectors and do plots in MATLAB.

General vector manipulation

At the MATLAB prompt, enter the following commands and note each result.

```
%This is a comment

x=5

x=10;        %The semicolon suppresses terminal output

x

z=0:1:10     %This creates an 11-element vector

z(1:3)       %Here are the first three elements of z

zd5=z/5      %Divide each component of z by a scalar
```

```
zmpi=z*pi    %Multiply each component by pi=3.14....

zp3=z+3      %Add 3 to each component

a=z(1:3)./[5 2 10]   %Divide elements 1:3 of z by 5,2,10 respectively

b=z(2:4).*[3 7 5]    %Multiply elements 2,3,4 by 3,7,5 respectively

z2=z.^2      %Square each component of the vector

zhalf=sqrt(z)  %Take the square root of each component

zsin=sin(z)    %Take the sin of each component of z (in radians)

zexp=exp(z)    %Take e to the power of each component of z
```

Plotting

Suppose you are testing a circuit element by applying voltages 1, 2, 3 and 4 V at times of 0, 2, 4, and 6 s, while reading current values of 0.5, 1, 1.5, and 2 A at each time point. You want to use the voltages and currents to characterize the circuit element.

Make a plot of v(t) after storing the following code into an m-file. Open a new m-file by using the New option from the File menu pull down in MATLAB. Type the following code into the window that pops up.

```
v=1:4;
i=0.5:0.5:2;
t=0:2:6;
plot(t,v)
xlabel('time')
ylabel('voltage')
title('Voltage vs. Time')
```

Create a directory called C:\MATLABR11\work\yourAlphaNumber, where yourAlphaNumber is your alpha number, e.g. m029334. Save your m-file in the directory your just created using the name **voft.m**. You need to add the directory to the MATLAB path prior to running your m-file. To run the code you type **voft** at the MATLAB prompt.

If you are in the computer lab in ML2, ML3, or C-5 and want to make sure you don't lose the m-files you create, you will need to save these files on a floppy disk.

Make a plot of power vs. time by typing,

```
p=v.*i;
plot(t,p),xlabel('time'),ylabel('power'),title('Power vs. Time');
```

Plotting as specific functions of time

Now suppose that the current is a function of time as given by

$$i(t) = \exp(-t / 10)$$

To plot this for time between 0 and 20 sec do the following:

```
clear          %To get rid of all the previous variable definitions
t=0:.1:20;     %Make the step size small enough so that the plot looks smooth
i=exp(-t/10);
plot(t,i),xlabel('time'),ylabel('current'),title('Current vs. time');
```

Student Exercises:

1. You've measured currents of 0, 1.1, 2.05, 3.08, and 3.95 mA through a circuit element when voltages of 0, 1.00, 2.01, 3.00, and 4.05 V were applied. Plot voltage versus current. (Note that 1.1 mA may be typed as **1.1e-3** in MATLAB.)

2. Plot $v(t)$=20V sin($2\pi 100t$) for t between 0 and 20 ms. (You want the curve to look like a sine wave so make sure you choose enough points between 0 and 20 ms.)

3. Add a legend to this plot.

4. Overlay $15e^{-100t}$ on the existing figure.

5. Update the legend.

6. If you print this figure you will not be able to distinguish one line from the other. Type **help plot** and experiment with different techniques to help solve this problem.

7. Type **grid**.

8. Type **grid** again.

9. Can you move the legend?

10. Investigate **axis**.

11. Investigate **semilogx**.

12. Investigate **semilogy**.

13. Plot $e^{-t}\cos(2\pi t)$ for t = 0 to 5 seconds.

14. Add e^{-t} to the existing figure as a dotted green line.

15. Add a legend, axes labels, and a title to the figure

16. Investigate the **rand** and **randn** functions.

17. Create a function that will generate N random zeros and ones (e.g. 0 1 0 1 0 1 0 0 1 1). Now take this vector and replicate each bit M times. For example, if M = 2, 0 becomes 0 0, and 1 becomes 1 1. So our 0 1 0 1 0 1 0 0 1 1 becomes 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 1 1 1 1. Your program should work for arbitrary values of N and M.

18. Investigate the **xcorr** function.

19. Investigate the **psd** function.

20. Investigate the **fft** function. Plot the results of an fft calculation.